Multi-body Segmentation and Motion Number Estimation via Over-Segmentation Detection

Guodong Pan and Kwan-Yee K. Wong

Department of Computer Science, The University of Hong Kong, Hong Kong

Abstract. This paper studies the problem of multi-body segmentation and motion number estimation. It is well known that motion number plays a critical role in the success of multi-body segmentation. Most of the existing methods exploit only motion affinity to segment and determine the number of motions. Motion number estimated in this way is often seriously affected by noise. In this paper, we recast the problem of multi-body segmentation and motion number estimation into an over-segmentation detection problem, and introduce three measures, namely loss of spatial locality (LSL), split ratio (SR) and cluster distance (CD), for over-segmentation detection. A hierarchical clustering method based on motion affinity is applied to split the motion clusters recursively until over-segmentation occurs. Over-segmentation is detected by Kernel Support Vector Machines trained under supervised learning using the above three measures. We leverage on Hopkins155 database to test our method and, with the same motion affinity measure, our method outperforms another state-of-the-art method. To the best of our knowledge, this paper is the first to tackle the problem of multi-body segmentation and motion number estimation from the perspective of over-segmentation detection.

1 Introduction

To reconstruct or understand a dynamic scene consisting of multiple moving objects observed by a static or moving camera, the trajectories of image features are often segmented using their motion affinity. Estimation of the motion number is critical to such a multi-body segmentation, and its failure often leads to a high error rate in the motion segmentation. In this paper, we refer to *motion number* as the number of independently moving objects in a scene.

Most of the existing works, if not all, exploit only motion affinity to segment and determine the number of motions. In the factorization method presented by Costeira and Kanade [1], the motion number was determined by sorting the shape interaction matrix and detecting blocks via minimizing the Frobenius norm of the shape interaction matrix subject to some physical constraints. This detection method suffers a lot from noisy data, especially when the noise level is high. Gear [2] converted the data matrix into an echelon form, and features of the same motion shared the same zero positions in the synthetic case. The motion number was then given by the number of different configurations. He also provided a bipartite graph model for real data with noise, and tried to explain

it with probabilistic models. Nevertheless, he admitted that real data was too complex to be explained by this model. Vidal et al. [3] presented the concept of multi-body fundamental matrix for the segmentation problem, and retrieved the motion number from the rank of the matrix of Veronese mapping of trajectories. It is a non-trivial problem to estimate the rank of a matrix with noise. This method also requires a minimum number of trajectories for each motion, which may not be practical. In [4], trajectories were clustered based on the distance of subspace using spectral clustering. In [5], the authors introduced the ordered residual metric, and clustered the trajectories also by spectral clustering. For the spectral clustering method in [6], the motion number was equivalent to the multiplicity of the zero eigenvalue of graph Laplacian, and the affinity matrix of trajectories was usually generated in such a manner as Normalized Cut [7]. The parameters of this model are quite influential, but are difficult to adjust for different applications. From the perspective of information theory, Ma et al. [8] modelled the problem via lossy data coding and compression, with the assumption that the mixed data were drawn from a mixture of Gaussian distributions. Given data to be compressed and a distortion criterion, the motion number and segmentation were obtained by minimizing the coding length. This method generalizes the problem but only considers data with mixtures of Gaussian distributions. [9] and [10] tackled the motion number estimation problem with a sampling method based on Torr's extension of Schwarz' BIC approximation [11]. Recent work [12] applied the Dirichlet Process Mixture Models to the motion hypotheses, and obtained the motion number when the process converged. However, with a median scale of disturbance and noise, the converged state was unsteady. [13] focused on the change of motion number in video and proposed a method based on an outlier detection approach. Most of the methods above determine the motion number only from the motion information, except [9] and [10] which used a local sampling scheme [14].

There is no doubt that motion affinity is a key factor for motion number estimation. However, this is by no means the only factor that matters. In this paper, we recast the problem of multi-body segmentation and motion number estimation into an over-segmentation detection problem, and introduce three measures, namely *loss of spatial locality* (LSL), *split ratio* (SR) and *cluster distance* (CD), to detect the occurrence of over-segmentation. A hierarchical clustering method based on an improved ordered residual metric is applied to split the motion clusters recursively until over-segmentation occurs. Supervised learning is employed to train Kernel Support Vector Machines using the above three measures motion affinity measure, our method outperforms another state-of-the-art method. To the best of our knowledge, this paper is the first to tackle the problem of multi-body segmentation and motion number estimation from the perspective of over-segmentation detection.

The rest of paper is organized as follows. Section 2 states our problem statement. Section 3 introduces the proposed measures for over-segmentation detection. The hierarchical clustering method and classifiers for over-segmentation

detection are described in Section 4. In Section 5, experiments and comparisons are presented. Finally conclusion and future work are discussed in the Section 6.

2 Problem Statement

Suppose several rigid objects are moving independently in a scene with different 3D motions, and a video camera is used to observe them. Feature points of the objects and the background are tracked through the video sequence. The problem of multi-body segmentation is to find the number of rigid motions and group the trajectories according to their motion affinity. Motion affinity refers to the degree to which motions share similar rotation and translation in 3D space. In this paper, we focus on objects in rigid motions and only consider the case when different moving objects have different 3D motions. We assume that all features are visible and tracked throughout the video sequence.

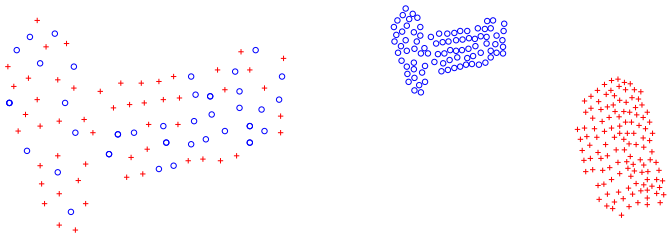
3 Measures for Over-Segmentation Detection

In this paper, the motion number is estimated by a recursive splitting approach. An initial motion cluster containing all the trajectories is recursively split into smaller clusters until over-segmentation occurs. When the recursion stops, the number of the resulting motion clusters simply gives the motion number. In the following subsections, we will introduce three measures for over-segmentation detection.

3.1 Loss of Spatial Locality

Assume that the moving objects are not transparent. Feature points of the same motion often scatter locally unless occlusion exists. Without occlusion, if two sets of features segmented into two different motions overlap, these features are likely being over-segmented. An example is shown in Fig. 1, where plus and circle marks denote features segmented into two different motions. The segmentation in Fig. 1(b) is more reasonable than that in Fig. 1(a) because there is no overlapping of the features, and hence shape integrity is not violated. Obviously, the overlapping of features in different motion clusters is a strong cue for over-segmentation.

Based on the above observation, we introduce a measure, namely *loss of spatial locality* (LSL), for over-segmentation detection. Given a motion affinity measure, a dataset can be divided into a number of motion clusters. For each element in a cluster, the number of its neighbors belonging to a different cluster is counted. LSL is defined as the total sum of such a number for all elements in all clusters, and it provides a measure for the degree of overlapping. If a feature set of the same motion is segmented into two motion clusters with a perfect motion affinity measure, every feature will have a probability of 0.5 to be selected into either cluster. A high LSL score would therefore mean the clusters



(a) (b)

Fig. 1. Plus and circle marks denote features segmented into two different motions. (a) Overlapping of features segmented into different motions suggests the occurrence of over-segmentation. (b) There is no overlapping of the features and hence shape integrity is not violated.

are highly overlapped and vice verse. For simplicity, K-Nearest Neighbor is used in determining the neighbors of a feature, and LSL is formulated as

$$LSL = \frac{1}{FN} \sum_{f=1}^{F} \sum_{i=1}^{N} G(x_{i,f}, k), \tag{1}$$

where F is the number of frames in the sequence, N is the number of feature points, $x_{i,f}$ is the i-th feature point in the f-th frame, $G(x, k)$ is the number of neighbor points belonging to a different cluster within the k-nearest point set of $x_{i,f}$ in term of image distance.

3.2 Split Ratio and Cluster Distance

Over-segmentation can also occur when there is no overlapping of feature sets. This can happen when the motion affinity measure is too sensitive which segments features on a rigid object into non-overlapping but adjacent motion clusters (see Fig. 2). For example, consider a car translating and rotating at a road junction. Motion affinity between features in the front (at the back) of the car would often score higher than those between the front and the back of the car. Consequently, features in the front of the car would often be segmented into one motion, and those at the back would be segmented into another motion. Obviously, LSL cannot detect this type of over-segmentation. Nonetheless, human can perceive such features sharing one single motion because (1) these non-overlapping clusters are relatively close to each other, and (2) they share similar motions. Based on these observations, two further measures, namely *split ratio* (SR) and *cluster distance* (CD), are introduced. SR is defined as the ratio of the smallest image distance between features in separate clusters to the largest one. It provides a measure for the distance between two non-overlapping clusters with respect to their sizes. Over-segmentation would produce a low SR score.

CD is defined as the distance between two cluster centers in the motion space. It measures how similar the motions of the two clusters are. Over-segmentation would produce a low CD score.

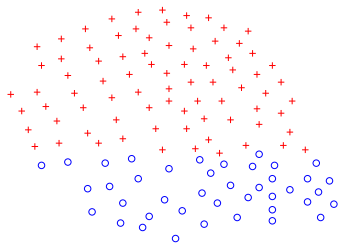


Fig. 2. Over-segmentation can also occur when there is no overlapping of feature sets. This can happen when the motion affinity measure is too sensitive which segments features on a rigid object into two non-overlapping but adjacent motion clusters.

4 Hierarchical Clustering with Supervised Classifiers for Over-Segmentation Detection

As mentioned before, the problem of multi-body segmentation is recast into an over-segmentation detection problem. A hierarchical clustering approach is adopted to recursively split the motion clusters until over-segmentation occurs. Initially, all trajectories are considered as one single motion cluster. An improved Ordered Residual metric is employed to split each motion cluster in two smaller clusters. This corresponds to building a binary tree in which the root node contains all the trajectories. Each split will produce two child nodes, the union of which is their parent node. After each split, classifiers trained under supervised learning are used to detect the occurrence of over-segmentation based on the previously introduced measures, namely loss of spatiality locality (LSL), split ratio (SR) and cluster distance (CD). If over-segmentation is detected in the split at a particular motion cluster, its child nodes will be removed from the binary tree and further splitting of its child clusters will be prohibited. Alg. 1 summarizes the algorithm of the proposed hierarchical clustering method. The improved Ordered Residual metric used for clustering and the classifiers used for over-segmentation detection will be described in detail in the following subsections.

Algorithm 1 Algorithm of the hierarchical clustering method.

```
Track image features to produce the trajectory data W;
Estimate the motion affinity K between each trajectory using Dual Pass Ordered
Residual method;
Dimension reduction: Project K onto the 4-D subspace corresponding to the 4 largest
singular values and get a 4-D point set D;
Create an empty queue Q and add a node R containing D to it;
Create an empty binary tree T and add R as the root node;
while Q not empty do
   Retrieve a node N from Q;
   Split the point set in N into two clusters by K-means;
   Compute LSL, SR and CD for the two child clusters;
   Assign the values of LSL, SR and CD to N;
   Use classifiers to decide if over-segmentation occurs;
   if over-segmentation not occurs then
      Add two new nodes containing the new clusters into Q;
      Add the two new nodes as child nodes of N in T;
   end if
end while
The number of clusters (motions) is given by the number of leaf nodes in T.
```

4.1 Dual Pass Ordered Residual Method

Several motion affinity measures have been mentioned in Section 1, such as shape interaction matrix [1], Local Subspace Affinity [4], and Ordered Residual [5]. Among these measures, the Ordered Residual method strongly interests us since it provides a more robust statistic estimation of motion affinity. In this paper, we propose an improved version of this method called *Dual Pass Ordered Residual method*, which is computational more efficient than the original method proposed in [5]. As its name suggests, the proposed method consists of two passes. In the first pass, we follow [5] in the way that a sufficient number of trajectory sets are randomly drawn to generate a hypothesis set, and the affinity matrix is computed. In the second pass, we fully exploit the information retrieved from the first pass by decomposition of the affinity matrix to obtain the nearest k neighbor of each trajectory in the motion space. For each trajectory, we obtain a refined hypothesis of the subspace by decomposition of the trajectories in the k neighbors instead of those selected randomly in the whole trajectory space. The number of hypotheses is independent of the size of the sampling, and we can obtain a satisfactory motion affinity matrix within two passes.

4.2 Classifiers for Over-Segmentation Detection

Although three measures for over-segmentation detection have been introduced in Section 3, it is still difficult to find a simple function relating them to make a decision on the occurrence of over-segmentation. Furthermore, over-segmentation

is more or less a subjective perception, with different people giving different opinions. Hence, a machine learning approach is adopted in this paper to learn the decision function.

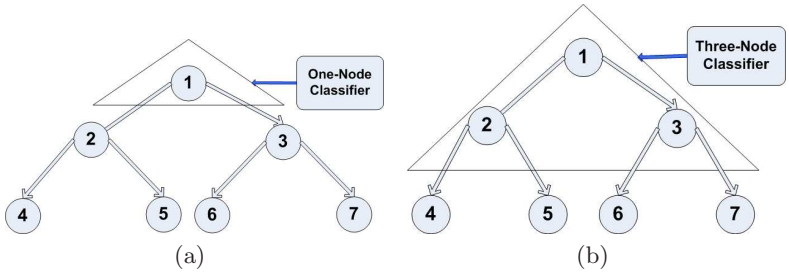


Fig. 3. A single-node structure for the root node and a triple-node structure for the non-root node

Each cluster node in the binary tree is associated with three features, namely *loss of spatial locality* (LSL), *split ratio* (SR) and *cluster distance* (CD), computed from its child nodes. A single-node structure and a triple-node structure are designed for classifying the split of a root node and a non-root node respectively. The single-node structure contains only one single node (see Fig. 3(a)), and is used for determining whether to split the root node or not based its associated features (i.e., LSL, SR and CD). The triple-node structure contains three nodes, including the node under consideration, its parent node, and its sibling node (see Fig. 3(b)), and is used for determining whether to split a non-root node or not based on the features of all three nodes (i.e., nine values in total). A classifier is trained for each type of structures respectively.

In the training stage, Kernel Support Vector Machines (SVM) with radial basis function [15] are trained under supervision. Observations are the features of the structures associated with each node, and labels are the decisions of whether to split or not. Observation collection includes two stages: dataset selection from the database as a training set and feature extraction. For dataset selection, we exploit two methods of cross-validation, namely K-fold cross validation and Hold-out cross validation, to evaluate the performance as the volume of the training set decreases. K-fold cross validation partitions the database into k folds, and uses $k-1$ portions as the training set and the rest for testing. For Hold-out cross validation, a portion of data will be hold out for testing and the rest will be used as training data. Both methods are applied because we want to find out the least portion of data needed to train the classifiers while keeping the performance. For each validation method, we train several SVMs and select the classifier giving the best performance. Feature extraction is carried out by the hierarchical clustering method introduced in the previous subsection, but without over-segmentation detection. A decision is labelled when a new structure appears.

5 Experiments

We leveraged on the benchmark of Hopkins155 [16] for experiments. Our method was applied to various real dynamic scenes with two to five motions, with both rigid and articulated motions. There are 119 two-motion examples, 35 three-motion examples and 1 five-motion example. To demonstrate the effectiveness of the proposed measures for over-segmentation detection, we compared our method with the spectral clustering method presented in [5]. To ensure a fair comparison, both methods used the same motion affinity metric as described in [5]. We have not compared our method with some other methods such as [9] and [10] because the performance of such methods heavily depends on the implementations.

To cluster the motions, we first computed the motion affinity as described in [5], centered the kernel matrix K and obtained its projection points P_4 in the 4-D subspace by eigen-value decomposition $K = RDR^T$ and $P_4 = D_4^{\frac{1}{2}} R(:, 1:4)^T$, where D_4 is the 4×4 diagonal block of D associated with the largest 4 eigenvalues, and $R(:, 1:4)$ consists of the 4 columns of R associated with the largest 4 eigenvalues. K-Means method was then used to cluster P_4 into two groups. This was done once in the testing stage but repeated eight times for the training stage to find the correct clustering. We computed LSL, SR and CD from the two groups for over-segmentation detection. The neighbor number for LSL was chosen to be 1 since we found any number within the range $[1, \ldots, 5]$ would give similar performance. For K-fold method, we trained k SVMs and selected the one with the best performance as our classifier. With Hold-out cross validation method, for each fixed portion, we repeated $1/portion$ times, each time trained one SVM and selected the one with best performance.

Table 1. Error rates for K-fold cross-validation.

FoldNumber	2	3	4	5	6	7	8
Overall	18.7%	16.1%	14.2%	17.4%	14.8%	15.5%	15.5%
TwoMotion	0%	0.8%	0%	0%	0%	0%	0%
ThreeMotion	77.1%	68.6%	60.0%	74.3%	62.3%	65.7%	65.7%
FiveMotion	100%	100%	100%	100%	100%	100%	100%

Table 2. Error rates for Hold-out cross-validation.

PortionForTest	95%	90%	85%	80%	75%	70%	65%	60%	55%
Overall	21.3%	21.9%	21.3%	21.3%	21.3%	19.4%	18.1%	18.1%	15.5%
TwoMotion	0%	0%	0%	0.8%	0%	0%	0%	0%	0%
ThreeMotion	91.4%	94.3%	91.4%	88.6%	91.4%	82.9%	77.1%	77.1%	65.7%
FiveMotion	100%	100%	100%	100%	100%	100%	100%	100%	100%

The error rate with K-fold is listed in Table 1. Overall error rate is defined by the ratio of the number of erroneously estimated examples to the total number in the database. Error rate of each motion number is also listed for analysis. We summarize the results of Hold-out in Table 2. From the tables, we can see our method did well in two-motion case but was not satisfactory for the three-motion and five-motion cases for both cross-validation methods. We also notice that the error rate of two-motion case in Hold-out was not very sensitive to the number of training samples. For example, the error rate associated with the case using 5% of data for training is the same with those using more training data. However, the error rate of three-motion case decreases as training data increase from 5% to 45%, which may indicates that there may be an insufficient number of three-motion and five-motion samples in the training set.

Table 3 below copies the results shown in Table 2 of [5] for ease of reference.

Table 3. Error Rates for [5]

Database	Hopkins 155
Overall	36.63%
TwoMotions	32.63%
ThreeMotions	50.34%

With benefit from the features for over-segmentation detection, our method outperforms [5] in most cases. For two-motion case, our method can virtually achieve an error rate of 0%. For three-motion case, the result of [5] is a little better than ours. One possible reason for the poor performance of our method is that the number of SVM parameter for three-motion and five-motions case is larger than that of the two-motion case, while the number of samples for the former in the database is much less that that of the later. The database hence provides an insufficient training set for the more-motion case.

6 Conclusion and Future Work

In this paper, we recast the problem of multi-body segmentation and motion number estimation into an over-segmentation detection problem. The main contributions of our work are (1) the introduction of three measures, namely loss of spatial locality, split ratio and cluster distance, for over-segmentation detection; (2) the introduction of the Dual Pass Order Residual method for computing motion affinity; (3) the introduction of a hierarchical clustering method for multi-body segmentation with a supervised learning approach for over-segmentation detection. We leverage on Hopkins155 database to test our method and, with the same motion affinity metric, our method outperforms another state-of-the-art method. To the best of our knowledge, this paper is the first to tackle the problem of multi-body segmentation and motion number estimation from the perspective of over-segmentation detection. In the future, more exploration should

be focused on the structures and features of over-segmentation that determine complex decision trees, such as a classifier structure for more than two motions.

References

1. Costeira, J.P., Kanade, T.: A multibody factorization method for independently moving objects. International Journal of Computer Vision **29** (1998) 159–179
2. Gear, C.: Multibody grouping from motion images. International Journal of Computer Vision **29** (1998) 133–150
3. Vidal, R., Ma, Y., Soatto, S., Sastry, S.: Two-view multibody structure from motion. International Journal of Computer Vision **68** (2006) 7–25
4. Yan, J., Pollefeys, M.: A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In: European Conference on Computer Vision. (2006) 94–106
5. Chin, T.J., Wang, H., Suter, D.: The ordered residual kernel for robust motion subspace clustering. In: Neural Information Processing Systems. (2009)
6. Luxburg, U.V.: A tutorial on spectral clustering. Technical report, Max Planck Institute for Biological Cybernetics (2007)
7. Shi, J., Malik, J.: Normalized cuts and image segmentation. In: Computer Vision and Pattern Recognition. (1997) 395–416
8. Ma, Y., Derksen, H., Hong, W., Wright, J.: Segmentation of multivariate mixed data via lossy data coding and compression. IEEE Transactions on Pattern Analysis and Machine Intelligence **29** (2007) 1546–1562
9. Schindler, K., Suter, D.: Two-view multibody structure-and-motion with outliers through model selection. Pattern Analysis and Machine Intelligence, IEEE Transactions on **28** (2006) 983 –995
10. Schindler, K., Suter, D., Wang, H.: A model-selection framework for multibody structure-and-motion of image sequences. International Journal of Computer Vision **79** (2008) 159–177
11. Bab-Hadiashar, A., Suter, D.: Chapter 6. In: Data segmentation and model selection for computer vision. Springer (2000) 143 – 178
12. Jian, Y.D., Chen, C.S.: Two-view motion segmentation with model selection and outlier removal by ransac-enhanced dirichlet process mixture models. International Journal of Computer Vision **88** (2010) 489–501
13. Ozden, K.E., Schindler, K., Gool, L.V.: Multibody structure-from-motion in practice. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 1134–1141
14. Schindler, K., Suter, D.: Two-view multibody structure-and-motion with outliers. In: Computer Vision and Pattern Recognition, IEEE Computer Society Conference on. Volume 2., Los Alamitos, CA, USA, IEEE Computer Society (2005) 676–683
15. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
16. Tron, R., Vidal, R.: A benchmark for the comparison of 3-d motion segmentation algorithms. In: IEEE Conference on Computer Vision and Pattern Recognition, 2007. (2007) 1 –8